

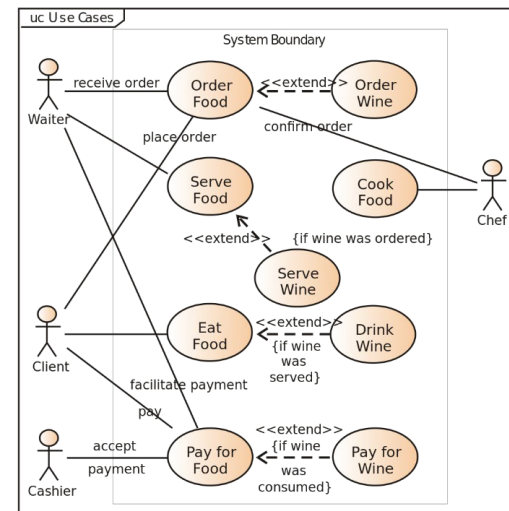
Agile Modeling

Different modeling techniques that are used to help establish the shared vision

Should be lightweight or “barely sufficient”

Agile Modeling

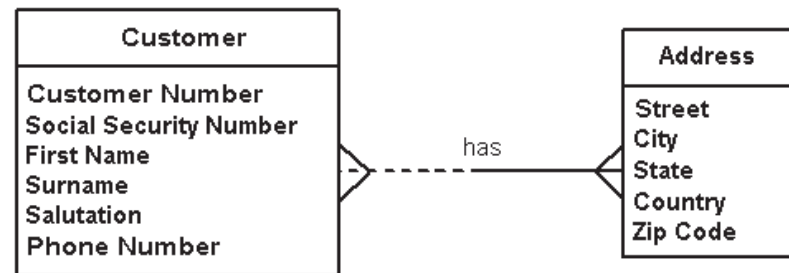
- Use case diagrams
 - Visually shows how users would use an application



https://en.wikipedia.org/wiki/Use_case_diagram

Agile Modeling

- Data models
 - How the data are structured in tables and their relationships



Copyright 2002-2006 Scott W. Ambler

<http://www.agiledata.org/essays/dataModeling101.html>

Agile Modeling

- Screen designs
- Simple screen shots

The image shows two hand-drawn wireframes for software screens. The left screen is titled 'Student Information' and contains fields for 'Student Number: 787-507-234', 'First Name: Scott', 'Middle: C', 'Surname: P', 'Salutation: Mr.', and 'Date first edit: June 11 2003'. Below these is a table for 'Semesters' with columns for 'Semester', 'Term', 'Date', and 'Status'. The table contains three rows of data. At the bottom are buttons for 'Add', 'Edit', 'Transcript', and 'Close'. The right screen is titled 'All a server' and contains fields for 'Server Number: CSC-8' and 'Name: C'. Below these is a table for 'Results' with columns for 'Server', 'Term', 'Grade', and 'Status'. The table contains three rows of data. Below the table is a text area for 'Course description' and a 'Close' button at the bottom right.

Student Number:	787-507-234		
First Name:	Scott		
Middle:	C		
Surname:	P		
Salutation:	Mr.		
Date first edit:	June 11 2003		
Semesters:			
Semester	Term	Date	Status
CSC 100 Intro to CS	Fall 2003	A+	Passed
CSC 200 Intro to AI	Fall 2003	A	Passed
CSC 300 Advanced AI	Spring 2004		Enrolled

Server Number:	CSC-8		
Name:	C		
Results			
Server	Term	Grade	Status
CSC 250 Agile Design	Fall 2003	4	Success
CSC 300 Agile Exp	Spring 2004	17	Success
CSC 300 Agile Design	Spring 2004	18	Success

<http://agilemodeling.com/artifacts/uiPrototype.htm>

Wireframes

Wireframes

- Quick mock-up of product
- “low-fidelity prototyping”
- Clarify what “done” looks like
- Validate approach prior to execution

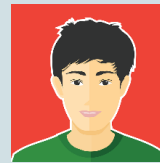
Personas

Personas

- Quick guides or reminders of key stakeholders and interests
 - Provide description of users
 - Be grounded in reality
 - Be goal-oriented, specific, and relevant
 - Be tangible and actionable
 - Generate focus
- Help team focus on valuable features to users

Personas

Name: Andrew Jones– Certified Accountant



Description:

Andrew has been an Accountant for over 10 years and has worked at many large accounting firms.

He likes to be organized and get his work done on time.

Value:

Andrew would like to ensure all company bills are paid on time while using online auto payments.

He would like to ensure customers are reminded automatically of outstanding balances.

He is looking to print the receivables and payable reports on a weekly basis to check on bills and invoices.

Communicating with Stakeholders

Face to face communication

Two-way communication

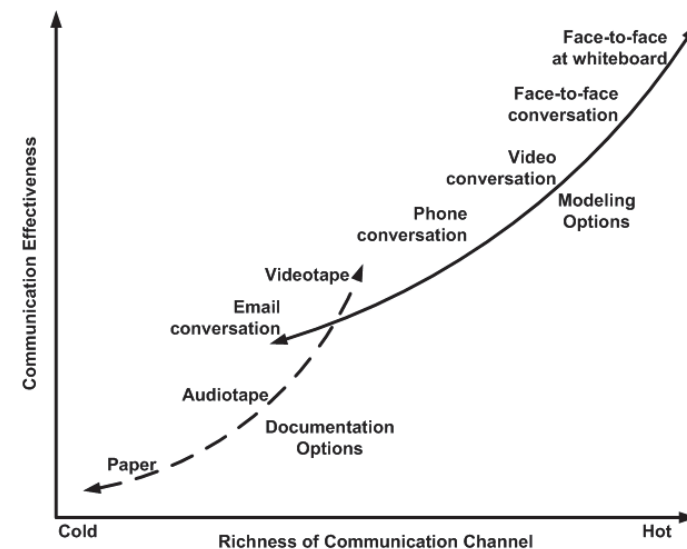
Knowledge sharing

Information Radiators

Social Media

Face-to-face Communication

Face to face communication



Copyright 2002-2005 Scott W. Ambler
Original Diagram Copyright 2002 Alistair Cockburn

<http://www.agilemodeling.com/essays/communication.htm>

Communicating with Stakeholders

Two-way communication

- Just don't ask for confirmation or concerns, but actually listen to what they have to say

Knowledge sharing

- Agile teams work closely with each other such as with pair-programming.
- Using Kanban boards or wireframes are ways to share information
- Use of low-tech tools like a whiteboard will allow all to see the work and understand it
- We must encourage it

Communicating with Stakeholders

Information Radiators

- Things that are highly visible
- Used to display information
- Usually includes chats, graphs and boards

Social Media

- Use to communicate
- Can include twitter or Instagram

Green Zone/Red Zone

Red Zone:

- Blames others for everything
- Responds defensively
- Feels threatened
- Triggers defensiveness
- Doesn't let go or forgive
- Uses shame and blame
- Focus on short-term advantage
- Doesn't seek or value feedback
- Sees conflict as a battle and only seeks to win
- Communicates high level of disapproval
- Sees others as the problem or enemy
- Does not listen effectively

Green Zone/Red Zone

Green Zone:

- Take responsibility
- Seeks to respond nondefensively
- Is not easily threatened psychologically
- Attempts to build success
- Uses persuasion rather than force
- Thinks both short and long term
- Welcomes feedback
- Sees conflict as a natural part of life
- Seeks excellence rather than victory
- Listens well

Using Workshops

Meeting when work gets done

Retrospectives are a type of workshops

Ways to make them more effective:

- Diverse groups has a larger perspective
- Use methods such as round-robin to ensure no one dominates
- Try to get everyone to participate in the first few minutes

User story workshops are where we write the user stories and keep stakeholders engage

Brainstorming

Brainstorming

- Quite Writing
 - Give people about 5 minutes to write down their ideas
- Round-Robin
 - Pass a token around to ensure everyone will speak
- Free-for-all
 - People shout out their thoughts. May only work in a supportive environment

Collaboration Games

Remember the future

Prune the product tree

Speedboat(Sailboat)

Remember the future

Ask stakeholders to imagine that an upcoming release was successfully and to look back

Gets a better understanding of how a stakeholder would define success

Outlines how we can accomplish that success for them

Prune the Product Tree

Draw a tree and ask stakeholders to add their features to it

Use stick notes to have them place new features on the tree

Group the features on the trunk

Features that are depending on other features would be higher up the tree

Lets everyone understand the priorities of development

Speedboat(Sailboat)

Draw a waterline and a boat moving

Explain the boat is moving toward the goals of the project

Ask them to use sticky notes to show what can make the boat move (wind) and what can stop it (anchors)

Allows stakeholders to identify threats and opportunities

Using Critical Soft Skills

Emotional intelligence

Negotiation

Active Listening

Facilitation

Conflict Resolution

Participatory Decision Models

Emotional intelligence

Our skill to identify, assess, and influence the emotions of ourselves and others around us

We need to recognize our own feeling

Then we can learn how to response to others and how they feel

Understand how we take care of ourselves will impact other around us

As an agile PM we have to know when team members are stuck, angry, or frustrated

Negotiation

This happens all throughout the project

Good negotiation will allow everyone to investigate the options and trade-offs

Most effective when interactions between people are positive and there are room for give and take

Active Listening

Level 1: Internal – how is it going to affect me

Level 2: Focused – put ourselves in the mind of the speaker

Level 3: Global – builds on level with body language

Facilitation

Run effective meeting and workshops.

Have the following:

- Goals
- Rules
- Timing
- Assisting

Conflict Resolution

All projects will have conflicts

While some level of conflicts are good, we need to ensure they don't become a "world war" where people are trying to destroy each other

Levels of conflict(1-5):

- Level 1: Problem to solve – sharing info
- Level 2: Disagreement – Personal Protection
- Level 3: Contest – Must win
- Level 4: Crusade – Protecting one's group
- Level 5: World War – Must destroy the other

Participatory Decision Models

Engage stakeholder in decision making process

- Simple voting
 - Vote “for” or “against” it
- Thumps up/down/sideways
 - People hold their thumps in a way of if the support it or not. Sideway is if they cannot make up their mind
- Fist of five
 - People how up finger based on they support the idea
 - 1 finger: total support – 5 finger: Stop against it

Team Performance

People Over Processes

Projects are done by people, not tools

- Agile manifesto: “Individuals and Interactions over processes and tools”

Focus on the people side of the project

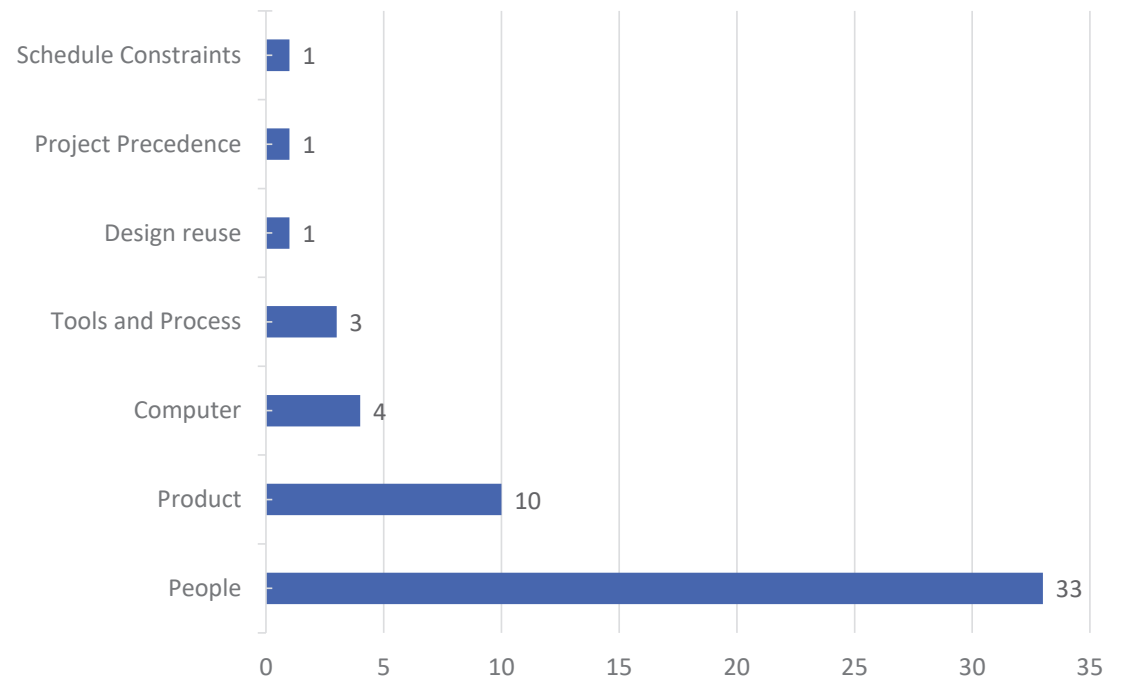
Projects are more about people management than tools management

People Over Processes

COCOMO

- Constructive Cost Model
- To determine correlation between project input variables and final cost to use to estimate future projects
- People factors has a score of 33...11 times more significant than tools and processes

COCOMO II



Development/Delivery Team

Group that build and test the increments of the product

- Build product in increments
- Update information radiators
- Self organize and directing
- Share progress by doing daily stand-up meetings
- Write acceptance tests
- Demo the completed product increments
- Holds retrospectives at the end of sprints
- Does release and sprint planning and estimations

Product Owner/Customer

Prioritizing the product features

Manage the product backlog ensuring its accurate and up to date

Ensures the team has a shared understanding of the backlog items

Defines the acceptance criteria

Provides the due dates for the releases

Attends planning meeting, reviews, and the retrospective.

Agile Project Manager (ScrumMaster/Coach)

Act as a servant leader

Help the team self-organize and direct themselves

Be a facilitator

Ensure the team plan is visible and the progress is known to the stakeholders

Act as a mentor and coach

Work with the product owner to manage the product backlog

Facilitates meeting

Ensure issues are solved

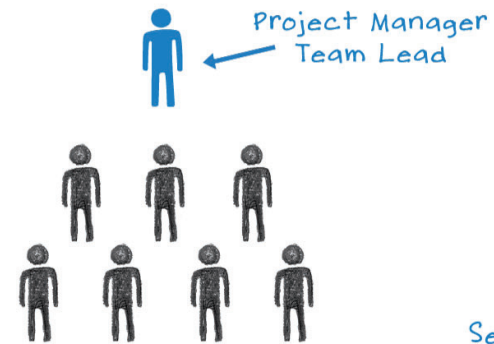
Building Teams

Self-Organizing

Self-Directing

Small teams with fewer than 12 members

Traditional Teams



Agile Teams



Generalizing Specialists

Have members that can do different tasks

Members skilled in more than one area

Share work reduce bottleneck

High-Performance Agile Teams

Have a shared vision

Realist goals

Fewer than 12 members

Have a sense of team identity

Provide strong leadership

Experiments (Have a safe place)

Establish safe environment for disagreement

Allows team members to build strong commitment to decisions

Encourage people to experiments with new methods

Leads to more engagement

Welcome Constructive Disagreement

Leads to better buy-in and decisions

Avoiding conflicts can lead to conflicts escalating

A safe place for disagreement leads to successful problem solving

Models of team development

Shu-Ha-Ri Model of Skill Mastery

- Shu- Obey,
- Ha – Moving away,
- Ri – finding individual paths

Dreyfus Model of Adult Skill Acquisition

- Novice, Advanced Beginner, Competent, Proficient, Expert

Tuckman's Five Stages of Team Development

1. **Forming:** team comes together and starts to get to know each other. There is not much conflict or communication.
2. **Storming:** team members start to have conflicts with each other. They start to learn of each other's ideas and may not agree with them. Most conflicts takes place in this stage.
3. **Norming:** the team members begin to agree with each other on the best methods to build the deliverables. Generally, everyone is coming to a consensus.
4. **Performing:** the team is performing well and is working without conflict.
5. **Adjourning:** In this stage, the project is completed and the team is reassigned.

Adaptive Leadership

Concept of adapting how we lead team based on specific circumstances and how mature team is in formation

Forming → Directing
Storming → Coaching
Norming → Supporting
Performing → Delegating
Adjourning

Training, Coaching, and Mentoring

Training

- Teaching of skills or knowledge

Coaching

- Process that helps a person develop and improve their skills

Mentoring

- More of a professional relationship that can fix issues on an as-needed basis

Help team stay on track, overcome issues, and continually improve skills

Individual level

Whole-team level

Team Spaces

Co-located Teams

Team Spaces

Osmotic Communication

Global and Cultural Diversity

Distributed teams

Co-Located Teams

All team member work together in the same location

Allows for face-to-face time and interaction

Should be within 33 feet of each other

No physical barriers

Sometimes a virtual co-location

Team Space

Lots of low-tech, high touch

- Whiteboards and task boards
- Sticky notes, flip charts
- Round table
- No barriers to face-to-face communication

Caves and Common

- Caves → space team members can retreat to individually
- Common → space team members can work as group

Osmotic Communication

- Information flows that occur as part of everyday conversations and questions
- 33 feet or 10 meters

Tacit Knowledge

- Information that is not written down; supported through collective group knowledge

Global and Cultural Diversity

Time Zones

Cultures

Native Languages

Styles of communications

Distributed Teams

At least one team member working off-site

Need to find ways to replicate co -location team benefits

Agile Tools

- Low-Tech, High-Touch Tools
- Digital Tools for distribute teams
 - Video conferencing
 - Interactive whiteboards
 - IM / VoIP
 - Virtual card walls
 - Web cams
 - Digital cams

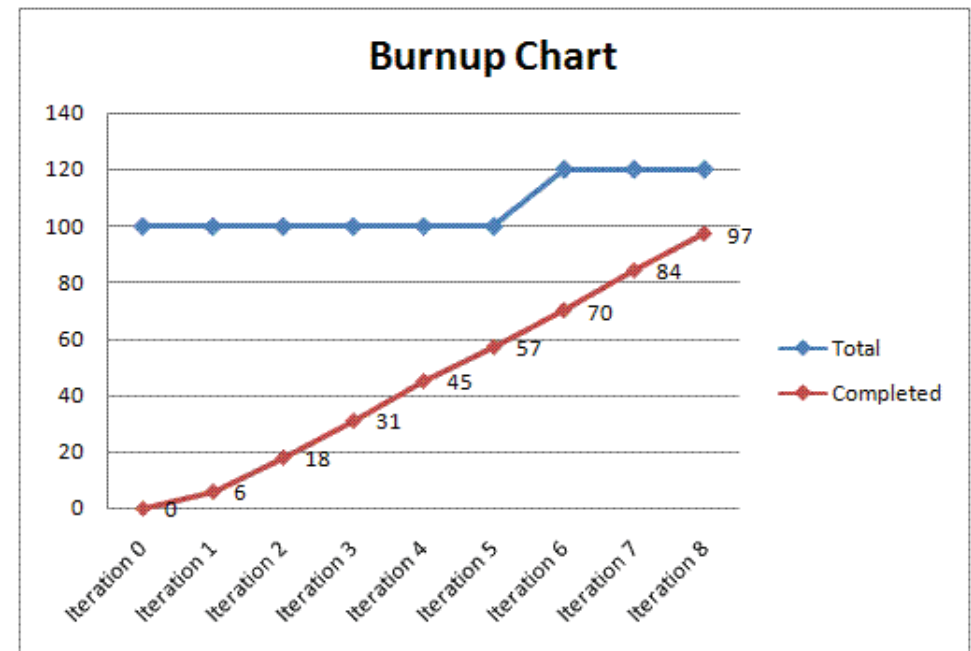
Tracking Team Performance

Burn Charts

- Burnup
- Burndown

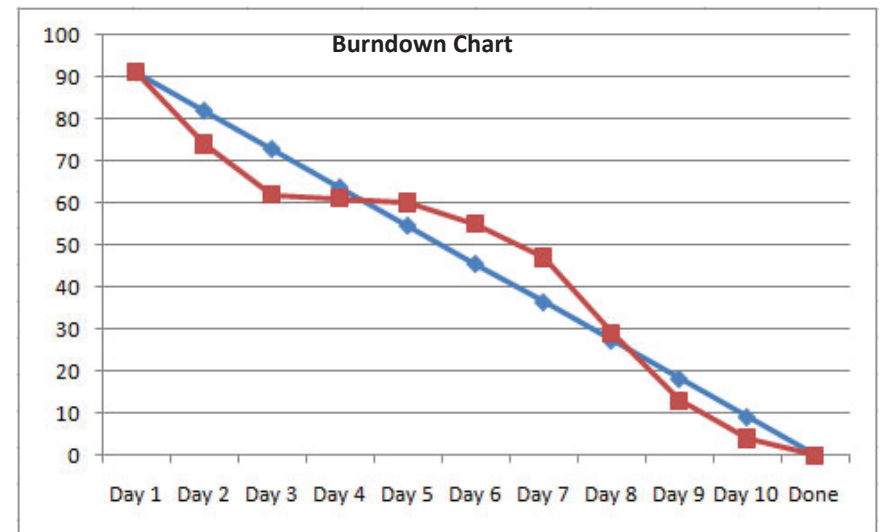
Velocity Charts

Burnup Chart



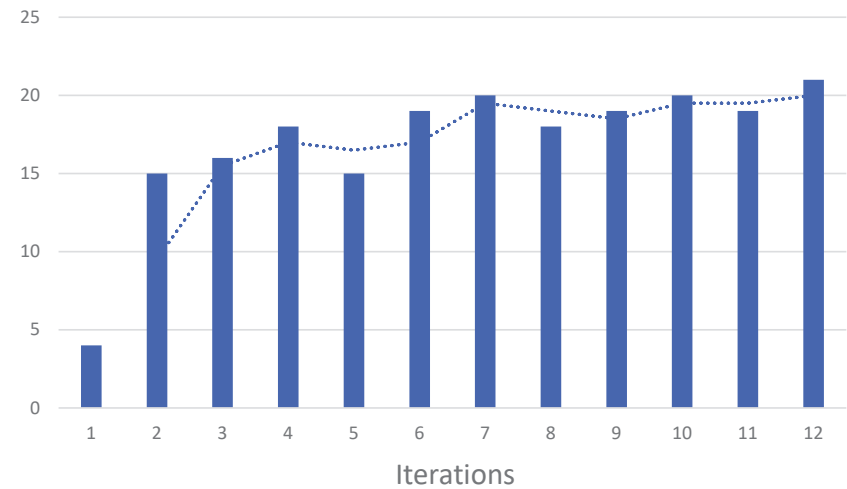
Work that has been done

Burndown Chart



Work that remains to be done

Velocity Charts



Show how the team is performing

Velocity Charts

If a team has complete 3 iterations with the average velocity of 18 points per iteration, how many iterations would it take to complete 250 points of work?

= $250/18$ = About 14 more iterations.

Adaptive Planning

Adaptive Planning

Planning is ongoing process

Multiple mechanisms to proactively update plan

Focus on value delivery and minimize nonvalue-adding work

Uncertainty drives need to replan

Frequently discover issues and experience high rates of change

Agile Plans

Agile planning varies from traditional planning

1. Trial and demonstration uncover true requirements, which then require replanning
2. Agile planning is less of an upfront effort, and instead is done more throughout the project
3. Midcourse adjustments are the norm

Principles of Agile Planning

1. Plan at multiple levels
2. Engage the team and the customer in planning
3. Manage expectations by frequently demonstrating progress
4. Tailor processes to the project's characteristics
5. Update the plan based on the project priorities
6. Ensure encompassing estimates that account for risk, distractions, and team availability
7. Use appropriate estimate ranges to reflect the level of uncertainty in the estimate
8. Base projections on completion rates
9. Factor in diversion and outside work

Progressive Elaboration

Adding more detail as information emerges

Includes:

- Plans
- Estimates
- Designs
- Test scenarios

Rolling wave planning: Planning at multiple points in time as data becomes available

Value-Base Analysis and Decomposition

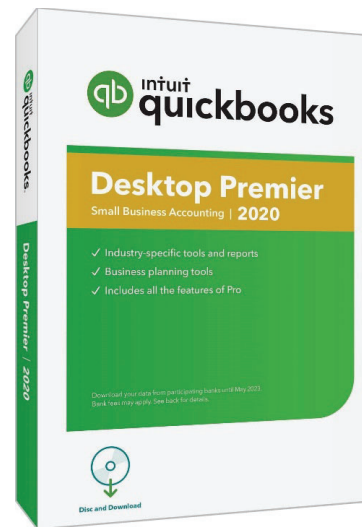
Assessing and prioritizing the business value of work items, and then plan accordingly.

Consider payback frequency and dependencies

Value -Based Decomposition

- Breaks down requirements and prioritized them
- Design the product box

Design the Product Box



“Coarse-Grained” Requirements

Keep Requirements “coarse” then progressively refine them

Helps keeps the overall design balanced

Delays decision on implementation until the “last responsible moment”

Timeboxing

Short, fixed-duration periods of time in which activities or work are undertaken

- If work is not completed within time period, move it to another timebox

Daily Stand-up – 15 minutes

Retrospectives – 2 hours

Sprints – 1-4 weeks

Beware of Parkinson's Law

- Work tends to expand to fill the time given

Agile Estimation

Knowledge of agile estimation theory & ability to perform simple agile estimating techniques

Why do we estimate?

- Determining which pieces of work can be done within a release or iteration

How are estimates created?

- By progressing through the stages planning.

How should estimates be stated?

- Should be started in ranges

When do we estimate?

- Throughout the project. More detail in the later parts of the project

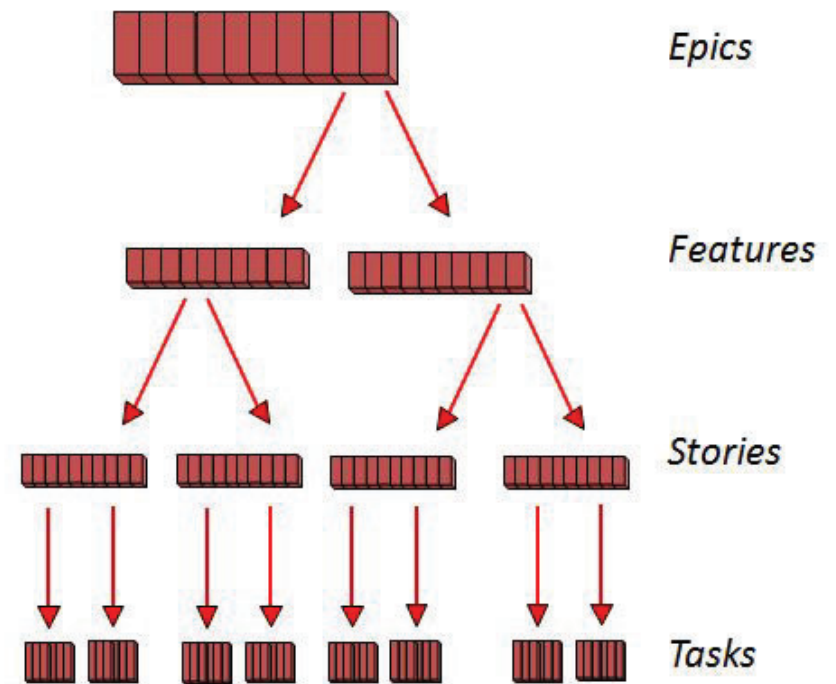
Who estimates?

- Team members will do their own estimates

Ideal Time

Refers to the time it would take to complete a given task assuming zero interruptions or unplanned problems

Decomposing Requirements



User Stories

User Stories / Backlogs

- Business functionality within a feature that involves 1-3 days of work.
- Acts as agreement between customers and development team
- Every requirement is user story
- Every story, including technical stories, has value
- Common structure of a user story

As a <user type>
I <want to/need, etc.> goal
So that <**value**>

User Story Example

“As an payroll clerk, I want to be able to view a report of all payroll taxes, so that I can pay them on time”

“As a sales person, I want to be able to see a current list of leads, so that I can call them back quickly”

“As student of this course, I want to be able to understand the requirements of the exam, so that I know if I qualify for it or not”

Three C's of Stories

Have users write the stories on index cards

No details, it's used to help converse

3 Cs:

- Card
- Conversation
- Confirmation

User Stories - INVEST

Effective user stories should be “INVEST”

Independent

- Should be independent so it can reprioritize

Negotiable

- Should allow for trade-off's based on cost and function

Valuable

- Should clearly state the value of it

Estimatable

- Should be able to estimate how long to complete

Small

- Stories should be between 4-40 hours of work

Testable

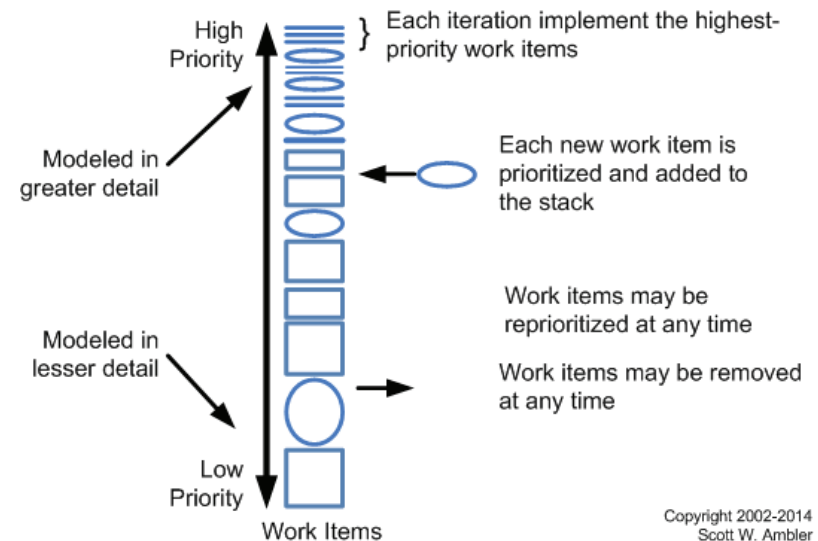
- Should be testable to ensure it will be accepted once completed

User Story Backlog (Product Backlog)

Prioritize Requirements

Refining (Grooming) Backlog

- Keeping the backlog updated and accurately prioritized



Copyright 2002-2014
Scott W. Ambler

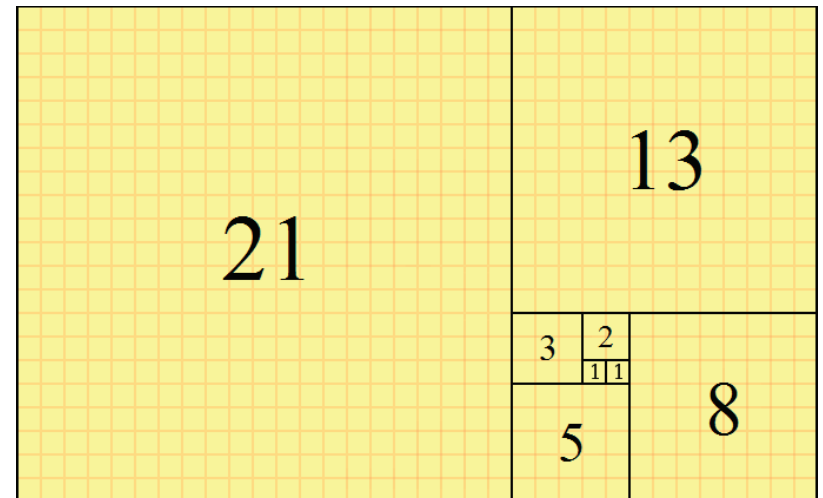
Relative Sizing and Story Points

Absolute estimates are difficult for humans to make

Estimates should be relative

Assign points to each story using a relative numbers

Fibonacci Sequence



https://en.wikipedia.org/wiki/Fibonacci_number

Fibonacci Sequence: 1, 2, 3, 5, 8, 13, 21

Guidelines for Using Story Points

Team should own the definition of their story points

Story point estimates should be all-inclusive

Point sizes should be relative

Complexity, work effort, and risk should all be included in the estimate

Affinity Estimating and T-Shirt Sizing

Affinity Estimating

- Group estimates into categories or collections

T-Shirt Sizing

- Place stories in sizes of t-shirts

Wideband Delphi

Wideband Delphi

- Group-based estimation approach
- Panel of experts, anonymously

It's used to prevent:

- Bandwagon effect
- HIPPO decision making (Highest-Paid Person's Opinion)
- Groupthink

Planning Poker

Advantages of Wideband Delphi

Fast, collaborative process

Uses cards with Fibonacci sequence

Story Maps

High-level planning tool

Stakeholders map out what the project priorities early in the planning

Serves as the “product roadmap”

Shows when features will be delivered and what is included in each release

Product Roadmap

Shows when features will be delivered and what is included in each release

Can convert the story map into a product roadmap

Types of Iterations

Iteration 0

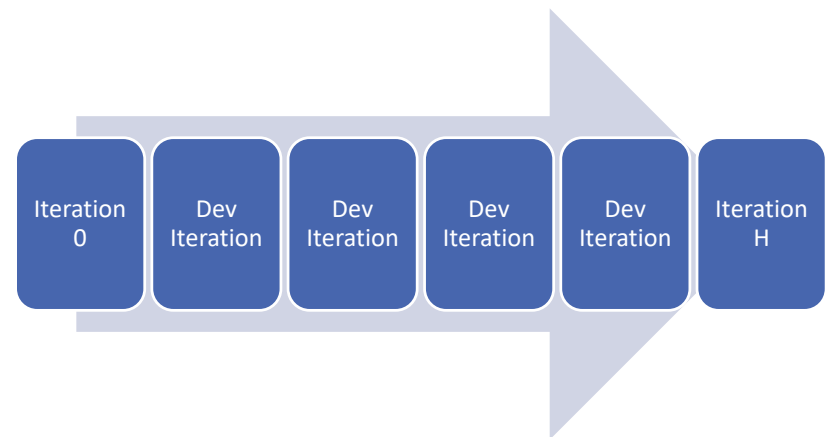
- Set the stage for development efforts
- Doesn't build anything

Development Iteration

- Build the product increment

Iteration H (hardening sprint or release)

- Done at the end to clean up codes or producing documentation



Spikes

- Architectural spike
 - Period of time dedicated to proof of concept
- Risk-Based Spike
 - Team investigate to reduce or eliminate risk

Iteration Planning

Meeting run by the delivery team.

Discuss the user stories in the backlog

Select the user stories for the iteration

Define the acceptance criteria

Break down the user stories into task

Estimate the task

Release Planning

Meeting with all stakeholders to determine which stories will be done in which iterations for the upcoming release.

Selecting the user stories for the release

- Using Velocity – points per iteration

Slicing the stories

- Breaking down stories that are too large to be completed in 1 iteration

Problem Detection and Resolution

Understand How Problems Happen

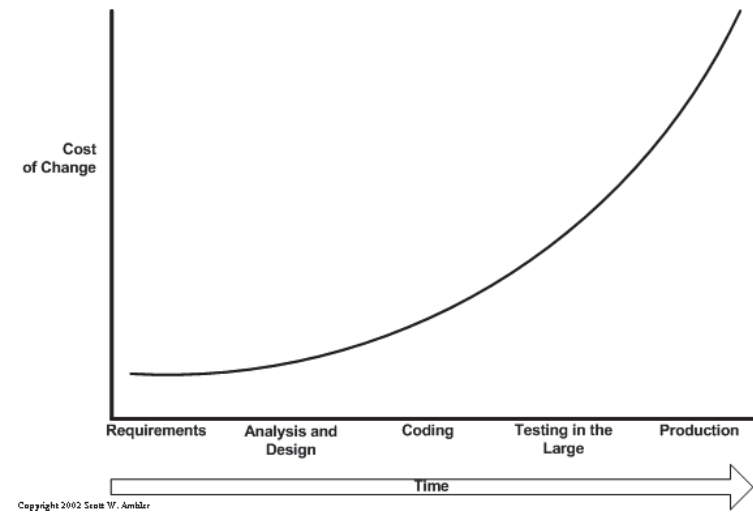
All projects will have problems

As a project is progressing the agile PM should expect issues to happen

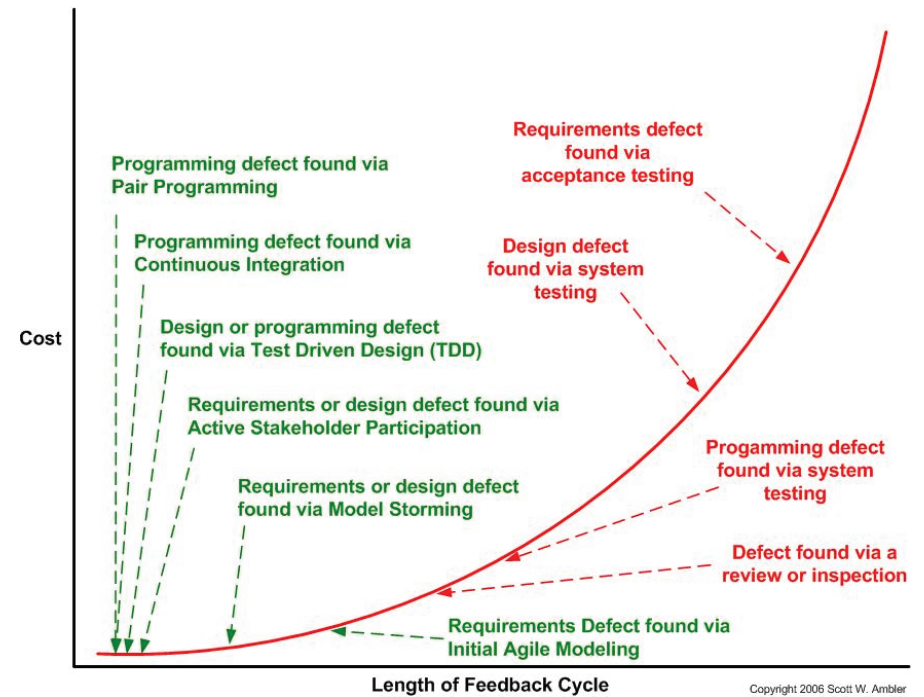
Over time issues can delay or change a project objectives

Cost of Change

Over time the cost of change will increase



Cost of Change



Technical Debt

Backlog of work caused by not doing regular cleanup

If not done will lead the increase cost of development and make it harder to implement changes

Refactoring is the solution

Failure Modes

Why do people Fail:

1. Making mistakes
2. Preferring to fail conservatively
3. Inventing rather than researching
4. Being creatures of habit
5. Being inconsistent

Success Modes

Why do we succeed:

1. Being good at looking around
2. Being able to learn
3. Being malleable
4. Taking pride in work

Success Strategies

Balance discipline with tolerance

Start with something concrete and tangible

Copy and alter

Watch and listen

Support both concentration and communication

Match work assignment with the person

Retain the best talent

Use rewards that preserve joy and combine rewards

Get feedback

Lead Time and Cycle Time

Lead/Cycle time

- Lead time: how long something takes to go through the entire process
- Cycle time: how long something takes to go through a part of the process. Part of lead time.

Cycle Time

- Measure of how long it takes to get things done
- Closely related to work in progress (WIP)
 - Excessive WIP is associated with several problems
 - Represents money invested with no return on investment yet
 - Hides bottlenecks in processes & masks efficiency issues
 - Represents risk in form of potential rework

Cycle Time

Long cycle times lead to increased amounts of WIP

$$\text{Cycle Time} = \frac{WIP}{Throughput}$$

Throughput: Amount of work that can done in a time period

Cycle Time Question

What would be the cycle time of feature A, if it requires 60 points of work and the team can complete 5 points per day?

= $60/5$ points per day = 12 days.

Defects

Longer defects are left, more expensive to fix

More work may have been built on top of bad design, resulting in more work to be undone

Later in development cycle, more stakeholders impacted by defect and more expensive to fix

Escaped Defects

- Defects that make it to the customer

Variance and Trend Analysis

Variance measure of how far apart things are (or vary)

Trend Analysis measure that provides insight into future issues

- Lagging Metrics provides information on something that has already happened
- Leading Metrics provides information on is or is about to occur

Control Limits

Help diagnose issues before issue occurs

Provide guidelines to operate within

Risk

Risk Adjusted Backlog

- Adjusting the backlog for risk
- Done after risk response

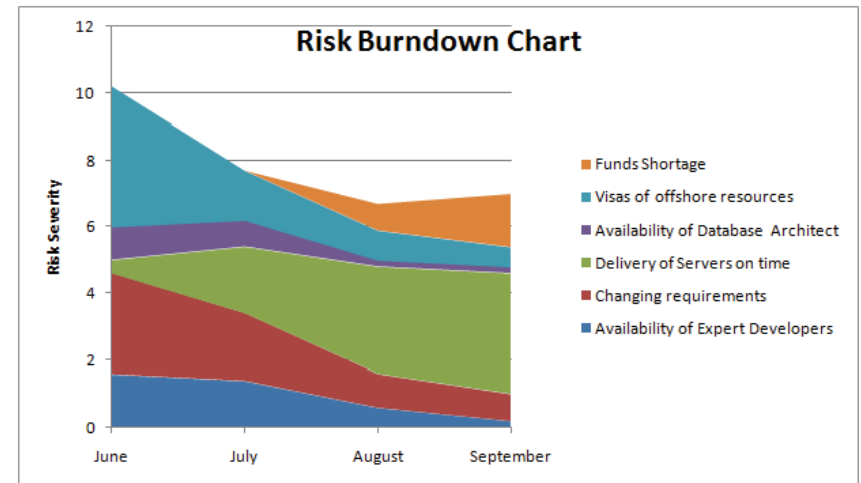
Expected Monetary Value = Impact(\$) x Probability(%)

Risk Severity

- Risk Probability x Risk Impact
- Uses a scale of numbers (E.g 1-5)

Risk

Risk Burndown Graphs



Solving Problems

Problem Solving as continuous improvement

Engage the team

Some problems can't be solved

Why Engaging the Team?

Team usually produces the best practical solutions

Benefits

- Get consensus from all members
- Gets a broad knowledge base
- Solutions are practical
- When ask people work hard to produce good ideas
- Asking someone for help shows confidence

Usage and Cautions

- Solve real problems
- Poor team cohesion
- Team and project changes
- Follow-Through

Continuous Improvement

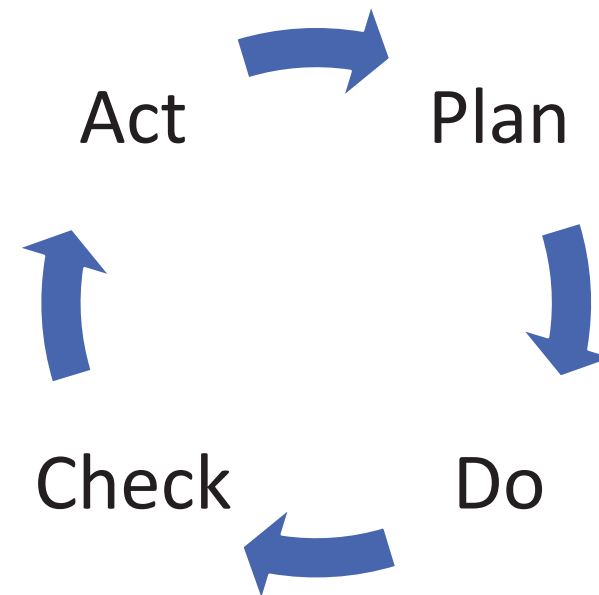
Kaizen

Kaizen is a process for continuous improvement
name after the Japanese word

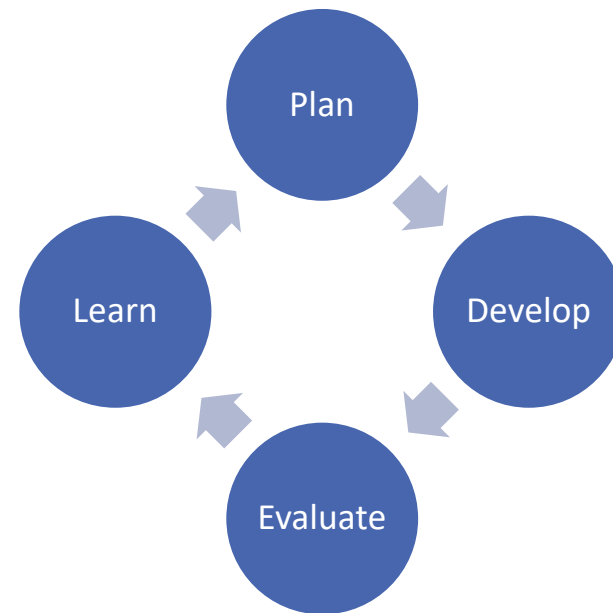
Focus on the team to implement small incremental
improvement

Usually follows the Plan-Do-Check-Act (PDCA) cycle
by Edwards Deming

PDCA



Agile Cycle



Process Analysis

Review and diagnose issues

Look for tailoring possibilities

Process Tailoring

Amend methodology to better fit project environment

Change things for good reason, not just for sake of change

Develop a hybrid

Value Stream Map

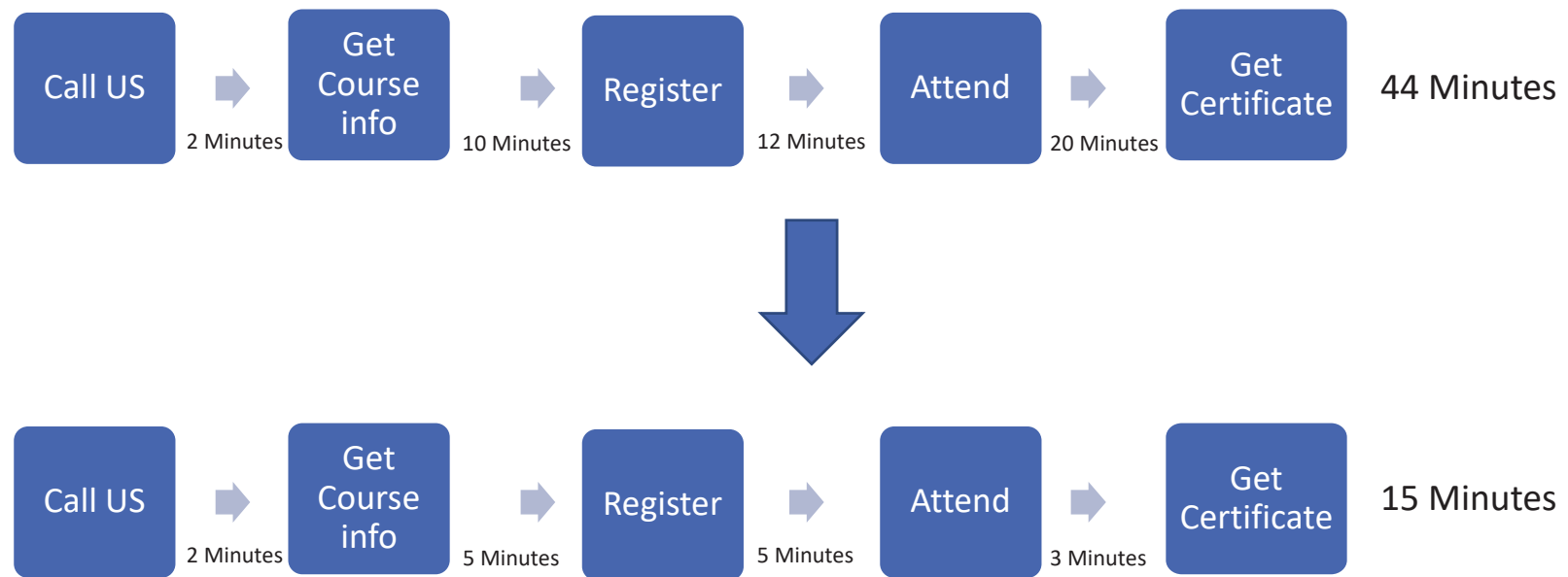
Optimize the flow of information or materials to complete a process

Reduce waste (waiting times) or unnecessary work

Steps to creating:

- Identify the product or service
- Create a value stream map
- Review to find waste
- Create a new map with the desire improvement
- Develop a roadmap to implement the fixes
- Plan to revisit it again

Value Stream Map Example



Pre-Mortems

Team meeting that looks at possible things that can cause failure during a project before they take place

Steps include:

- Think what the failures might be
- Create a list of reasons that can cause the failures
- Review the project plan to determine what can be done to reduce or remove the reasons for failure

Retrospectives

Special meeting that takes place after each iteration

Inspect and improve methods and team work

Offers immediate value

Should have a 2 hour time limit

Retrospectives Stages

About 2 Hours for a typical retrospective

1. Set Stage – 6 Minutes
2. Gather Data – 40 Minutes
3. Generate Insights – 25 Minutes
4. Decide What to Do – 20 Minutes
5. Close Retrospective – 20 Minutes

1. Set the Stage

Start of the retrospective

Help people to get focus

Encourage participation to ensure everyone start talking early

Outlining the approach and topics for discussion

Get people in mood for contributing information

Activities include:

- Check-In
- Focus On/Focus Off
- ESVP
 - People identify if they are an explorer, shopper, vacationer, or Prisoner

2. Gather Data

Create a picture of what happened during the sprint

Start to collect information to be used for improvement

Activities:

- Timeline
- Triple Nickels: break the team into 5 groups to spend 5 minutes collecting 5 ideas, 5 time
- Mad, Sad, Glad: what where the team emotion as the sprint was taking place

3. Generate Insights

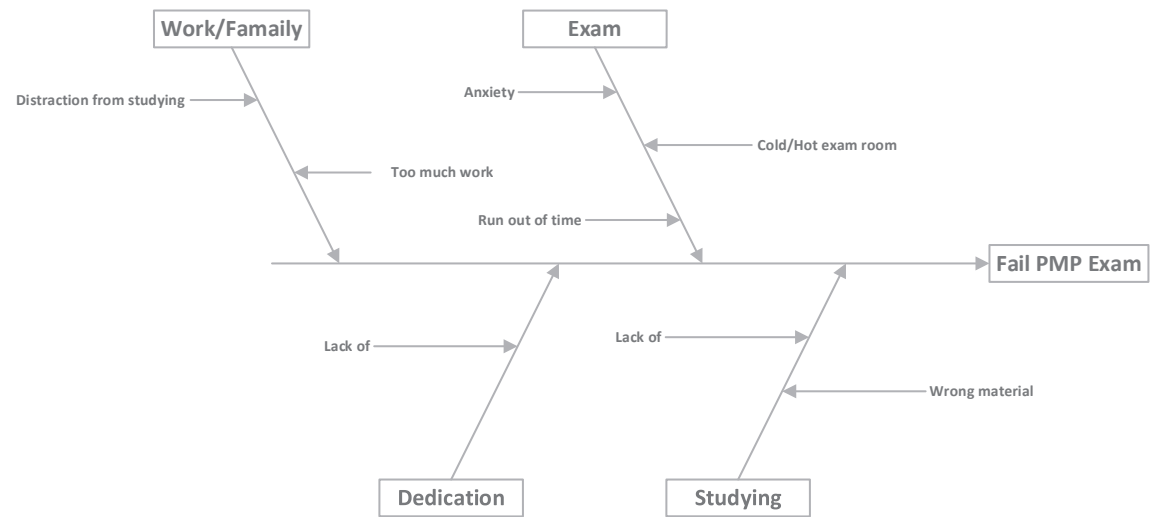
Analyze the data

Helps to understand what was found

Activities Include:

- Brainstorming
- Five Whys: asking why five times
- Fishbone analysis
- Prioritize with dots: use a dot voting technique

Fishbone Analysis



4. Decide what to do

Decide what to do about the problems that was found

How can we improve for the next iteration

Activates include:

- Short Subjects
- Smart Goals

Short Subjects

Team decides what actions to take in the next iteration:

- Start doing
- Stop doing
- Do more of
- Do less of

SMART Goals

Team sets goals that are SMART:

- **S**pecific
- **M**easurable
- **A**ttainable
- **R**elevant
- **T**imely

5. Close the Retrospective

Opportunity to reflect on what happened during the retrospective

Activities include:

- Plus/Delta: make two column of what the team will do more of and what to do less of

Team Self-Assessments

Uses to evaluate the team as a hold

Things to evaluate can include:

- Self-organization
- Empowered to make decisions
- Belief in vision and success
- Committed team
- Trust each other
- Constructive disagreement